FOODiE, Fortran Object oriented Ordinary Differential Equations integration library based on Abstract Calculus Pattern

Zaghi S.[a,1,*], Curcic M.[b,2], Rouson D.[c,3], Beekman I.[c,4]

[a] *CNR-INSEAN, Istituto Nazionale per Studi ed Esperienze di Architettura Navale, Via di Vallerano 139, Rome, Italy, 00128*
[b] *Ocean Sciences Rosenstiel School of Marine and Atmospheric Science, University of Miami, 4600 Rickenbacker Causeway Miami, FL 33149-1098 +1 305.421.4000*
[c] *Sourcery Institute 482 Michigan Ave., Berkeley, CA 94707*

Abstract

To be written.

Keywords: Ordinary Differential Equations (ODE), Partial Differential Equations (PDE), Object Oriented Programming (OOP), Abstract Calculus Pattern (ACP), Fortran,

PROGRAM SUMMARY

Manuscript Title: FOODiE, Fortran Object oriented Ordinary Differential Equations integration library based on Abstract Calculus Pattern
Authors: Zaghi, S., Curcic, M., Rouson, D., Beekman, I.
Program title: FOODiE
Journal Reference:
Catalogue identifier:
Licensing provisions: GNU General Public License (GPL) v3
Programming language: Fortran (standard 2008 or newer); developed and tested with GNU gfortran 5.2 or newer
Computer(s) for which the program has been designed: designed for shared-memory multi-cores workstations and for hybrid distributed/shared-memory supercomputers, but any computer system with a Fortran (2008+) compiler is suited
Operating system(s) for which the program has been designed: designed for POSIX architecture and tested on GNU/Linux one
RAM required to execute with typical data: bytes: $[1MB, 1GB] \times core$, simulation-dependent
Has the code been vectorised or parallelized?: the library is not aware of the parallel back-end, it providing a high-level models, but the provided tests suite shows parallel usage by means of MPI library and OpenMP paradigm
Number of processors used: tested up to 256
Supplementary material:
Keywords: ODE, PDE, OOP, ACP, Fortran
CPC Library Classification: 4.3 Differential Equations, 4.10 Interpolation, 12 Gases and Fluids
External routines/libraries used:
CPC Program Library subprograms used:
Nature of problem:
Numerical integration of (general) Ordinary Differential Equations system
Solution method:

*Corresponding author
Email addresses: stefano.zaghi@cnr.it (Zaghi S.), milan@orca.rsmas.miami.edu (Curcic M.), damian@sourceryinstitute.org (Rouson D.), izaak@izaakbeekman.com (Beekman I.)
[1] Ph. D., Aerospace Engineer, Research Scientist, Dept. of Computational Hydrodynamics at CNR-INSEAN.
[2] Ph.D. Meteorology and Physical Oceanography, Research Scientist, Dept. of Ocean Sciences Rosenstiel School of Marine and Atmospheric Science at University of Miami
[3] Ph.D. Mechanical Engineering, Founder and President Sourcery Institute and Sourcery, Inc.
[4] Graduate Research Assistant, Princeton/UMD CCROCCO LAB

Restrictions:
Unusual features:
Additional comments:
Running time:
References:

4.1. Oscillation equations test

$$\begin{aligned} U_t &= R(U) \\ U = \begin{bmatrix} x \\ y \end{bmatrix} \quad R(U) &= \begin{bmatrix} -fy \\ fx \end{bmatrix} \end{aligned} \tag{1}$$

where the frequency is chosen as $f = 10^4$. The ODE system 1 is completed by the following initial conditions:

$$\begin{aligned} x(t_0) &= 0 \\ y(t_0) &= 1 \end{aligned} \tag{2}$$

where $t_0 = 0$ is the initial time considered. The exact solution is:

$$\begin{aligned} x(t_0 + \Delta t) &= X_0 cos(f\Delta t) - y_0 sin(f\Delta t) \\ y(t_0 + \Delta t) &= X_0 sin(f\Delta t) + y_0 cos(f\Delta t) \end{aligned} \tag{3}$$

where Δt is an arbitrary time step.

4.1.1. Errors Analysis

For the analysis of the accuracy of each solvers implemented into FOODiE library, we have integrated the Oscillation equations 1 with different, decreasing time steps in the range $[5000, 2500, 1250, 625, 320, 100]$.

The error is estimated by the L2 norm of the difference between the exact (U_e) and the numerical ($U_{\Delta t}$) solutions for each time step:

$$\varepsilon(\Delta t) = ||U_e - U_{\Delta t}||_2 = \sqrt{\sum_{s=1}^{N_s} (U_e(t_0 + s * \Delta t) - U_{\Delta t}(t_0 + s * \Delta t))^2} \tag{4}$$

Using two pairs of subsequent-decreasing time steps solution is possible to estimate the order of accuracy of the solver employed computing the *observed order* of accuracy:

$$p = \frac{log10\left(\frac{\varepsilon(\Delta t_1)}{\varepsilon(\Delta t_2)}\right)}{log10\left(\frac{\Delta t_1}{\Delta t_2}\right)} \tag{5}$$

where $\frac{\Delta t_1}{\Delta t_2} > 1$.

Table 1 summarizes the errors analysis.

Adams-Bashforth.

Leapfrog.

Low Storage Runge-Kutta.

TVD/SSP Runge-Kutta.

Table 1: Oscillation test errors analysis

Solver	Time Step	Error	Observed Order of Accuracy
Adams-Bashforth 2 steps	5000	1565.886	/
	2500	34.608	5.500
	1250	11.117	1.638
	625	3.815	1.543
	320	1.391	1.508
	100	0.243	1.500
Adams-Bashforth 3 steps	5000	11.491	/
	2500	3.746	1.617
	1250	1.698	1.141
	625	0.769	1.142
	320	0.314	1.339
	100	0.059	1.438
leapfrog 2 steps	5000	21.456	/
	2500	7.099	1.596
	1250	2.705	1.392
	625	2.560	0.079
	320	2.222	0.212
	100	1.417	0.387
low storage Runge-Kutta 5 stages	5000	$1.715\ 10^{-1}$	/
	2500	$1.509\ 10^{-2}$	3.507
	1250	$1.331\ 10^{-3}$	3.503
	625	$1.175\ 10^{-4}$	3.501
	320	$1.128\ 10^{-5}$	3.500
	100	$1.925\ 10^{-7}$	3.500
TVD/SSP Runge-Kutta 2 stages	5000	44.954	/
	2500	12.626	1.832
	1250	4.287	1.558
	625	1.506	1.510
	320	0.551	1.502
	100	0.096	1.500
TVD/SSP Runge-Kutta 3 stages	5000	3.582	/
	2500	$7.350\ 10^{-1}$	2.285
	1250	$1.330\ 10^{-1}$	2.471
	625	$2.349\ 10^{-2}$	2.497
	320	$4.407\ 10^{-3}$	2.500
	100	$2.406\ 10^{-4}$	2.500
TVD/SSP Runge-Kutta 5 stages	5000	$1.976\ 10^{-1}$	/
	2500	$1.744\ 10^{-2}$	3.502
	1250	$1.539\ 10^{-3}$	3.502
	625	$1.361\ 10^{-4}$	3.499
	320	$1.333\ 10^{-5}$	3.470
	100	$7.294\ 10^{-7}$	2.498

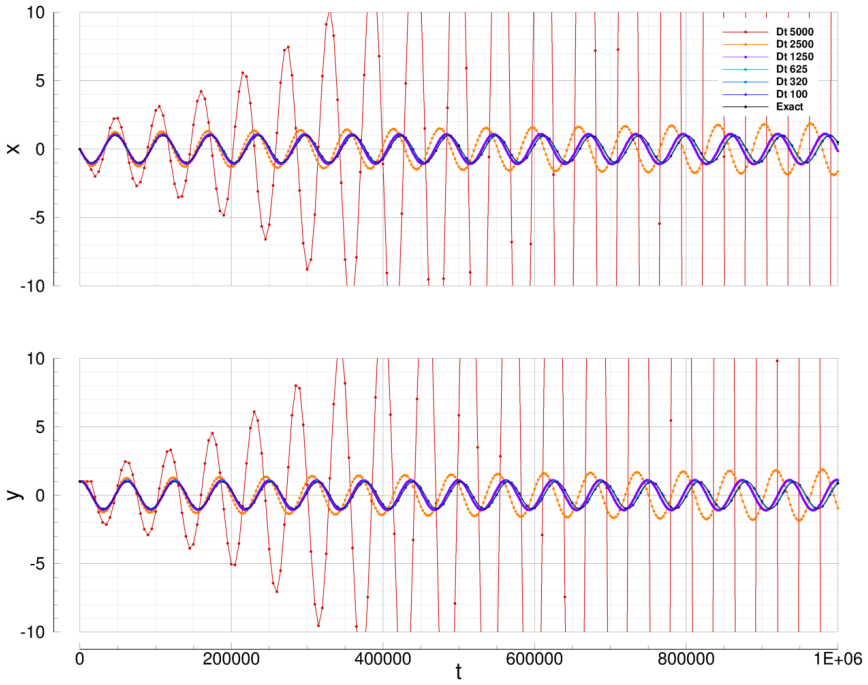

Figure 1: Oscillation equations solutions computed by means of Adams-Bashforth 2 steps solver

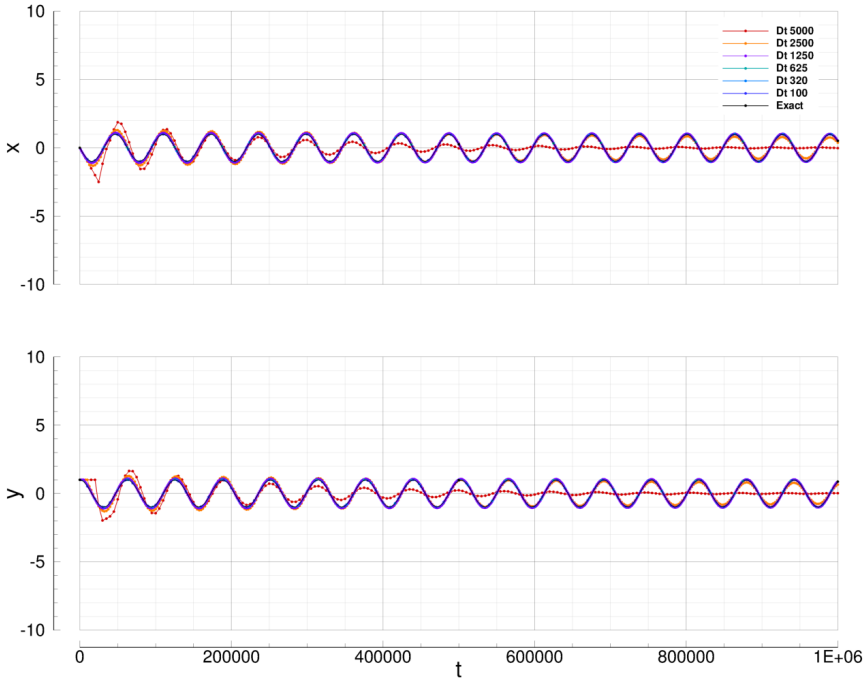

Figure 2: Oscillation equations solutions computed by means of Adams-Bashforth 3 steps solver

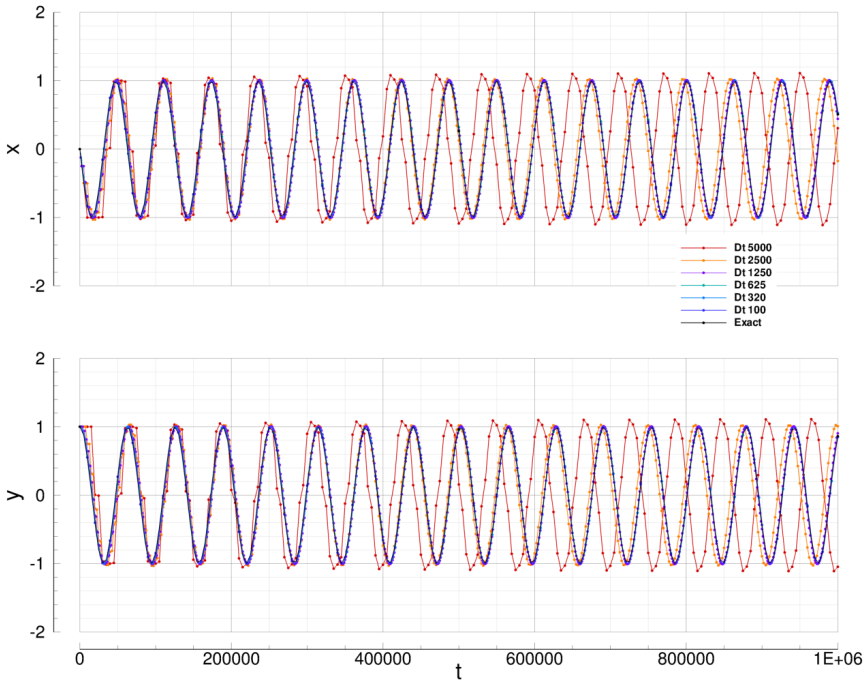

Figure 3: Oscillation equations solutions computed by means of leapfrog 2 steps solver

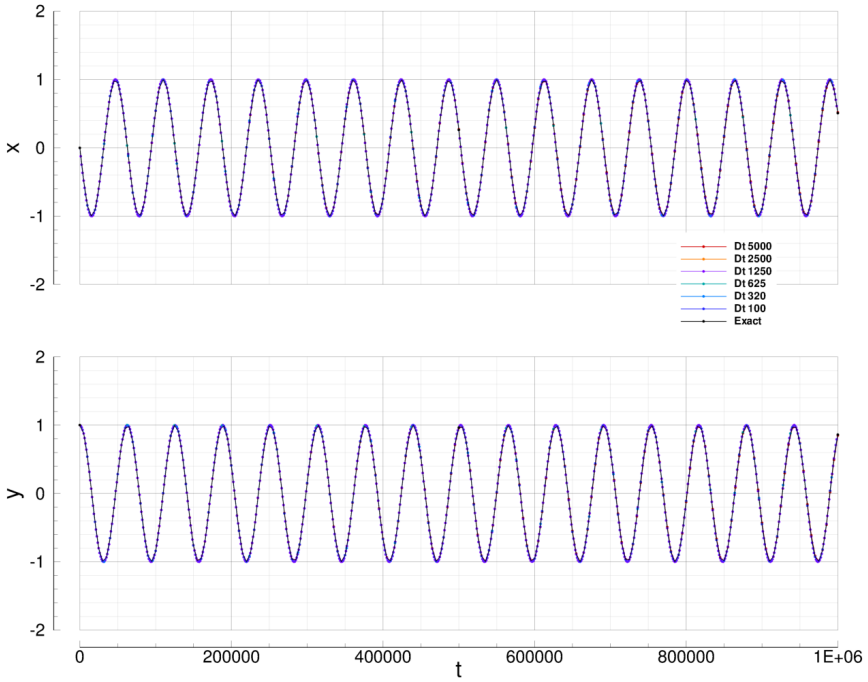

Figure 4: Oscillation equations solutions computed by means of low storage Runge-Kutta 5 stages solver

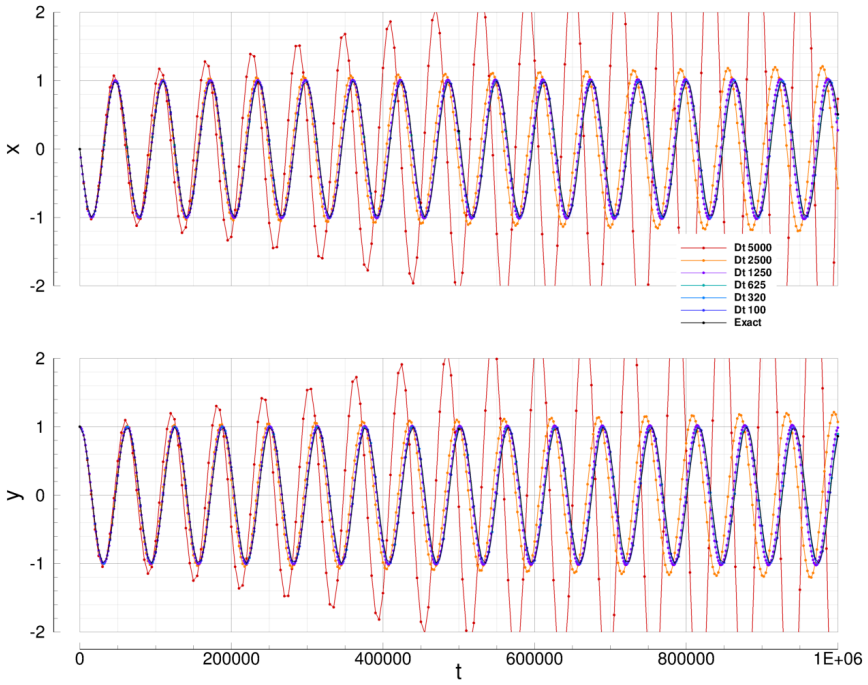

Figure 5: Oscillation equations solutions computed by means of TVD/SSP Runge-Kutta 2 stages solver

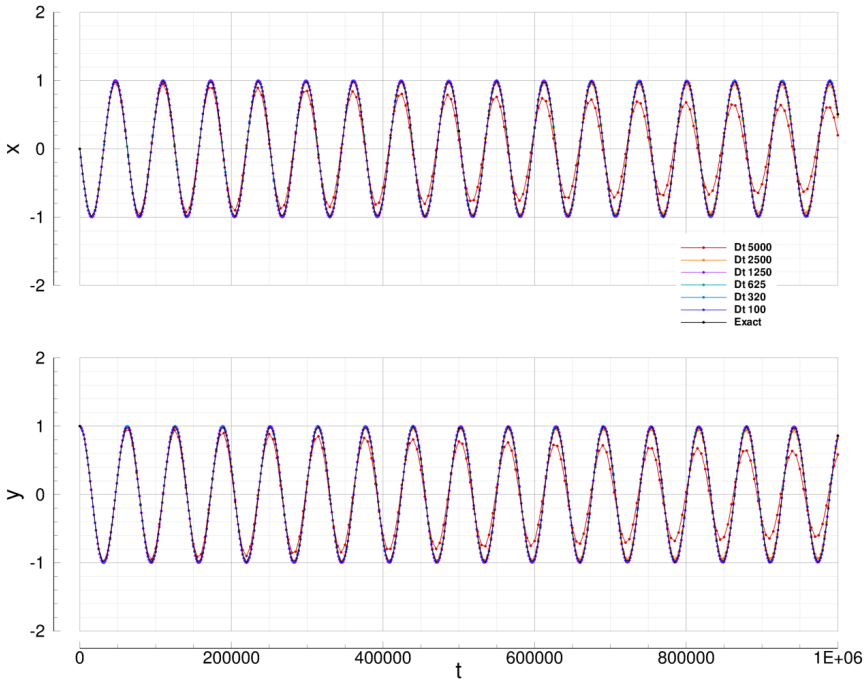

Figure 6: Oscillation equations solutions computed by means of TVD/SSP Runge-Kutta 3 stages solver

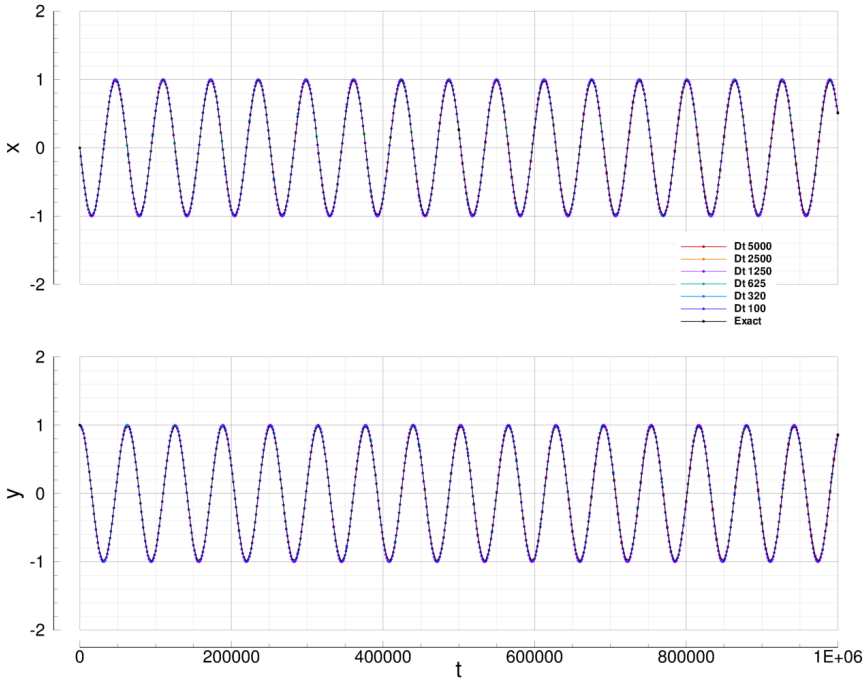

Figure 7: Oscillation equations solutions computed by means of TVD/SSP Runge-Kutta 5 stages solver